Received: June 25, 2019

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Robust Feature Extraction and Classification Based Automated Human Action Recognition System for Multiple Datasets

Jagadeesh Basavaiah[1]* **Chandrashekar Mohan Patil**[1]

[1] *Department of Electronics and Communication Engineering,*
Vidyavardhaka College of Engineering, Mysuru India
* Corresponding author's Email: jagadeesh.b@vvce.ac.in

Abstract: Action recognition schemes enable several intelligent machines to recognize human action by using daily life videos. In the last decades, recognizing human actions in the video sequences have been a challenging problem because of its real-world applications. Several action representation techniques have been done to improve action recognition performance. Approaches based on local features and classification are used for action representation, but it failed to capture temporal relationship between actions. In this work, Video Action Recognition (VAR) is assessed by using Weizmann, KTH and own datasets. Initially frames are extracted from input videos and the frames are resized. After the pre-processing, the object detection is done by Blob detection algorithm and tracking the object frame by frame is done using Kalman filter. The features are extracted from the moving object using feature extraction algorithms such as Bi-dimensional Empirical Mode Decomposition (BEMD), Scale Invariant Feature Transform (SIFT) and Discrete Wavelet Transform (DWT). These feature extraction techniques were applied on pre-processed frames to extract the efficient features from multi-scale images. Finally, the features are given to Convolution Neural Network (CNN) classifier for action recognition prediction. The proposed method is called as Hybrid Feature Extraction (HFE) and CNN used for VAR (HFE-CNN-VAR) method. The experimental results showed that the HFE-CNN-VAR method improved the accuracy in action classification. The classification accuracy is 99.01% for KTH dataset, 99.33% for Weizmann dataset and 90% for own dataset.

Keywords: Convolution neural network, Dimensional empirical mode decomposition, Discrete wavelet transform, Scale invariant feature transform, Video action recognition.

1. Introduction

The video sequences provide much significant information about actions compared to an image. The recognition of human-induced actions in videos has gained an important amount of interest in fields of the pattern recognition and computer vision [1], because of its increasingly large number of the applications in the areas of Human-Machine Interaction [HMI] [2], virtual reality, intelligent space, elderly care [3], robotics, and so on. In practical applications, Human Action Recognition (HAR) system is frequently affected by high intra-activities, camera view variation conditions and a large number of activities. The conventional methods based on various types of input data can be classified into three types such as depth based action recognition, RGB based action recognition, and skeleton action recognition [4, 5]. The major goal of the action recognition field is to provide the ability to identify human action in Real Life Videos (RLV) [6].

The HAR in the video has drawn increasing attention in modern computer vision studies [7, 8]. However, exact action recognition is complex because the challenges in the realistic scenarios may lead to the interactivities variation in similar action groups in terms of scale change and dynamic viewpoint [9]. Video action recognition is generally regarded as a classification issue; hence it is primary to design a network to learn effective representation of activities [10]. In this work, the proposed method consists of two processes: the training and testing process. In the training process, initially, frames are extracted from the input videos and resized, and colour frames in case of Weizmann and own dataset are converted to grayscale. After pre-processing, the feature extraction algorithms such as BEMD, SIFT, DWT are applied on the pre-processed frames to extract the efficient features from multi-scale images, which considered as a testing output. In the testing process, the object detection is done by blob detection to find the region, moving object was tracked frame by frame using Kalman filter based on time. Efficient features are extracted in tracking objects by BEMD, SIFT and DWT. Finally, the trained and tested values are given to CNN for action recognition prediction.

This research paper is composed of following sections: Section 2 presents an extensive survey of recent papers on HAR and classification techniques. Section 3 briefly described HFE-CNN-VAR methodology. Section 4 shows the comparative experimental result for the existing and HFE-CNN-VAR method. The conclusion is made in Section 5.

2. Literature survey

The researchers have suggested numerous video action recognition techniques for HAR in surveillance applications. A brief evaluation of few significant video action recognition is presented in this section.

Y. Zhao, H. Di, J. Zhang, Y. Lu, F. Lv, and Y. Li [11] proposed region-based mixture models used for HAR in the Low-Resolution Video (LRV). Here, the Layered Elastic Motion Tracking (LEMT) method was utilized for extracting a set of long term motion trajectories and long term common shape from every video sequence. An extracted trajectory is heavy thick compared with Sparse Interest Points (SIP). A hybrid feature representation used to combine both the shape and motion features. Region based mixture model utilized for action classification. The region-based mixture model encodes the spatial layout of features without any necessity of human region segmentation. The background noise is the major drawback of this work.

C. Zhang, Y. Tian, X. Guo, and J. Liu [12] proposed a Joint Semantic Preserving Action Attribute Learning (JSPAAL) method for HAR from depth videos, which is developed on a multi-stream Deep Neural Network (DNN). Furthermore, this work defines the idea to explore action attributes learned from the deep activities. The multiple stream DNNs rather than existing hand crafted low-level features are used to learn to deep activities. In this study, an undirected graph is used to model the difficult semantics among action attributes and integrated into the proposed method. Extensive experiments prove the proposed method is effective in the learning action attributes for depth videos, but the complexity of the model is very high.

In Zhang [13], proposed an improved trajectory-based human action recognition technique to capture discriminative temporal relationships. In this proposed method, an extract trajectory by tracking the detected Spatiotemporal interest points called as cuboid features with matching its SIFT on the consecutive frames. Next, the volume around the trajectories points was defined to represent human actions based on Bag of Words (BoW) model. At the final stage, Support Vector Machine (SVM) was used to classify human actions. But, an improvement of only 0.4 % was achieved when analysed on the KTH dataset over the dense trajectories method. It is not suitable for real time processing.

L. Yao, Y. Liu, and S. Huang [14] implemented a low-level feature based framework used for human activity recognition. In this work, the Spatio temporal information used among videos to recognize human actions. Initially, proposed a Spatio temporal bigraph based feature fusion to combine various features. In second, introduced a Spatio temporal video representation, which employs the Spatio-temporal distance between features to compute the distance visual words. Although, the early fusion of several features in KTH and Olympic datasets have obtained average accuracy, this effect is not so good for few classes: running and hand clapping.

Chou et al. [15] compared three practical, reliable and generic systems for multi-view video based action recognition, such as nearest neighbour classifier, Gaussian mixture model classifier, and nearest mean classifier. To define the various actions performed in various views, and view-invariant features were proposed to find multi-view action recognition. These features were obtained by extracting the holistic features from various temporal scales, which are modelled as points of interest that represent the global spatial-temporal distribution. However, these proposed approaches do not perform properly on a MuHAVi dataset.

In this research, the HFE-CNN-VAR method is implemented to overcome the above-mentioned limitations and to improve the performance of video action recognition.

3. Proposed methodology

The major goal of action recognition is to give the capability to the computer for recognizing human activities in standard as well as own dataset videos. In this research, video action recognition is

Received: June 25, 2019

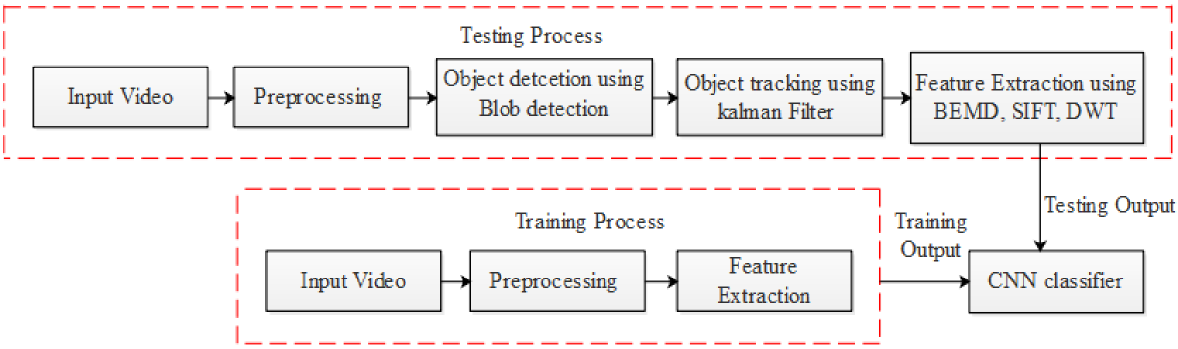


Figure.1 Training and testing process of the HFE-CNN-VAR method

implemented based on the training and testing process. The brief explanation of the training and testing process of HFE-CNN-VAR method is depicted in this section.

3.1 Training and testing process of HFE-CNN-VAR method for video action recognition

Fig.1 shows a block diagram of the HFE-CNN-VAR training and testing process, which is consist of image processing techniques such as pre-processing, object detection, object tracking, and feature extraction. Initially, the video sequences are collected from Weizmann, KTH and own database.

Frames are extracted from the video and the extracted frames are resized. In the first phase, video action is considered in particular dataset and the moving object is detected in each frame by blob detection. In the second phase, tracking of an object is done by Kalman filter to localize regions, points or features of the image frame by frame. In the third phase, the three types of feature extraction methods (BEMD, SHIFT, and DWT) are applied on the frame to extract features at multiple scales. The proposed HFE-CNN-VAR method is briefly explained as follows.

3.1.1. Object detection using blob detection scheme

After the pre-processing, human detection is done by blob detection scheme for obtaining a specific region of interest to perform human tracking in video. Recently, the blob detection has found increasingly popular because it employs interest points for significant baseline stereo matching and signalling the presence of informative image features used for appearance based object detection on the basis of local image statistics. The major aim of blob detection scheme is to find regions in a digital image that has various properties in terms of brightness and surrounding regions. Each blob region is stretched in vertical and horizontal directions until the whole blob is enclosed in a rectangular box. In this work, the blob detection system is based on adjacency pixels, boundary box and the centre of mass. The Kalman filter scheme is applied to blob detected region for frame by frame tracking.

3.1.2. Kalman filter tracking scheme for robust human (object) tracking

Object tracking approach is performed by analysing the object's position forms the previous information and verifying the existence of the object at the analysed position. Kalman filter evaluates a process by utilizing feedback control. This type of filter evaluates the procedure state at the same time and it obtains feedback in form of the noisy measurements. There are two types of the equations used for Kalman filters in terms of measurement update equations and time update equations. The time update based equations are responsible for projecting and forward the present state and the Error Covariance (EC) to obtain a prior estimate for the further time step. The measurements update based equations are responsible for feedback, which is utilized for incorporating a new measurement to achieve improved posterior estimate. The time update expression is given in Eq. (1).

$$Xpred_k = M \times X_{k-1} + N \times U_k + W_{k-1} \qquad (1)$$

$$Ppred_k = M \times P_{k-1} \times A^T + L \qquad (2)$$

Here, $Xpred_k$ is vector denoting predicted process state at time k, X is a four dimensional vector $(x, y\ dx\ dy)$, x and y denote coordinates of object's centre, dx and dy denote its velocity, X_{k-1} is a vector representing process state at time $k-1$, M is 4 ×4 process transition matrix of the form,

$$M = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

U_k is control vector and N relates optional control vector U_k into state space. W_{k-1} is process noise. In Eq. (2), $Ppred_k$ represents predicted EC at time k. The P_{k-1} is a matrix denoting EC in the state recognition at time $k-1$ and L is process noise covariance. After predicting the state $Xpred_k$ and its EC at time k utilizing the time update steps. Next, the kalman filter employs measurement approach to correct its evaluation during the measurement update steps.

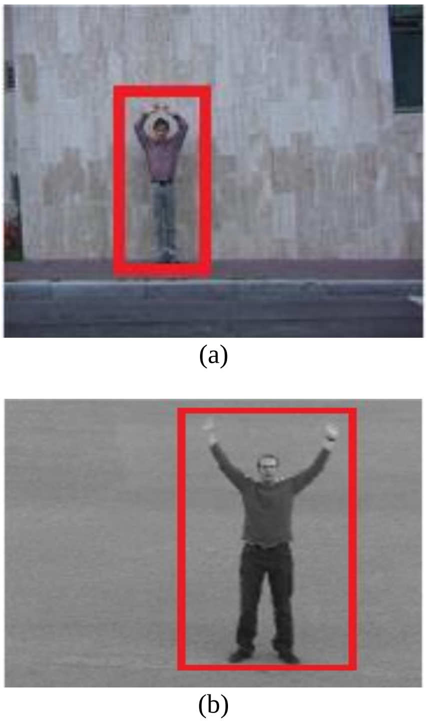
$$K_k = Xpred_k \times E^T \times \left(E \times Ppred_k \times E^T + F\right)^{-1} \quad (3)$$

$$X_k = Xpred_K + K_k * \left(Z_k - E \times Xpred_k\right) \quad (4)$$

$$P_k = (1 - K_k \times E) \times Ppred_k \quad (5)$$

In Eq. (3), K_k is Kalman filter gain, E is matrix converting state space into measurement space, and F is measurement noise covariance. In Eq. (4), X_k is the process of actual state. Using Kalman filter gain and the measurement Z_k process state can be updated. Here, Z_k is the most considered for x and y coordinates to target the objects in the frame.

The final step in the Kalman filter is to obtain the EC $Ppred_K$ from P_k which is expressed in Eq. (5). After each time pair and measurement pair, the process is repeated with previous posterior estimates employed to predict the new prior estimate. Fig. 2 (a) and (b) shows the sample tracking Hand 2 waving image in Weizmann and KTH dataset.



(a)

(b)

Figure.2 (a) Sample tracking Hand 2 waving image in Weizmann dataset and (b) Sample tracking Hand 2 waving image in KTH dataset

After the object tracking, the three different types of feature extractions techniques used for extracting the optimal features, which is briefly explained as follows.

3.1.3. Bi-dimensional empirical mode decomposition

Initially, the BEMD algorithm utilized to decompose the self-adaptive features of the original image, to obtain several Bi-dimensional Intrinsic Mode Function (BIMF) components. In this research the 2 Dimensional image has been represented by$f(x,y)$. The fundamental BEMD decomposition process is expressed in below seven steps.

Step1: The image under consideration is initialized. $r_0(x,y) = f(x,y), k = 1, (x,y)\epsilon[0, M-1] \times [0, N-1]$. Here, M and N are the number of ranks on the discrete frame plane.

Step 2: Initialize the parameters $h_{k,0}(x,y) = r_{k-1}(x,y)$ and, $l = 1$.

Step 3: Extract extrema points of $h_{k,l-1}(x,y)$.

Step 4: By cubic spline, interpolate between the local maxima and minima, to get two envelope surfaces $e_{max,l-1}(x,y)$ and $e_{min,l-1}(x,y)$ of $h_{k,l-1}(x,y)$.

Step 5: Compute the mean envelope surface in terms of these two envelop surfaces, it is given in Eq. (6).

$$e_{mean,l-1}(x,y) = \frac{1}{2}\left[e_{max,l-1}(x,y) + e_{min,l-1}(x,y)\right] \quad (6)$$

Step 6: Update the original signal and designate a new one for iteration, it is expressed by Eq. (7).

$$h_{l,k}(x,y) = k_{k,l-1}(x,y) - e_{mean,l-1}(x,y) \quad l \to l+1 \quad (7)$$

Step 7: Compute the standard deviation, which is expressed by Eq. (8).

$$SD = \sum_{x=0}^{X} \sum_{y=0}^{Y} \frac{\left|h_{k,l-1}(x,y) - h_{k,l}(x,y)\right|^2}{h_{k,l-1}^2(x,y)} \quad (8)$$

Step 8: Step (2) and (7) are repeated until the evaluated SD value is less than the pre-determined criterion. The $h_{k,l}(x,y)$ is considered as a representative of BIMF.

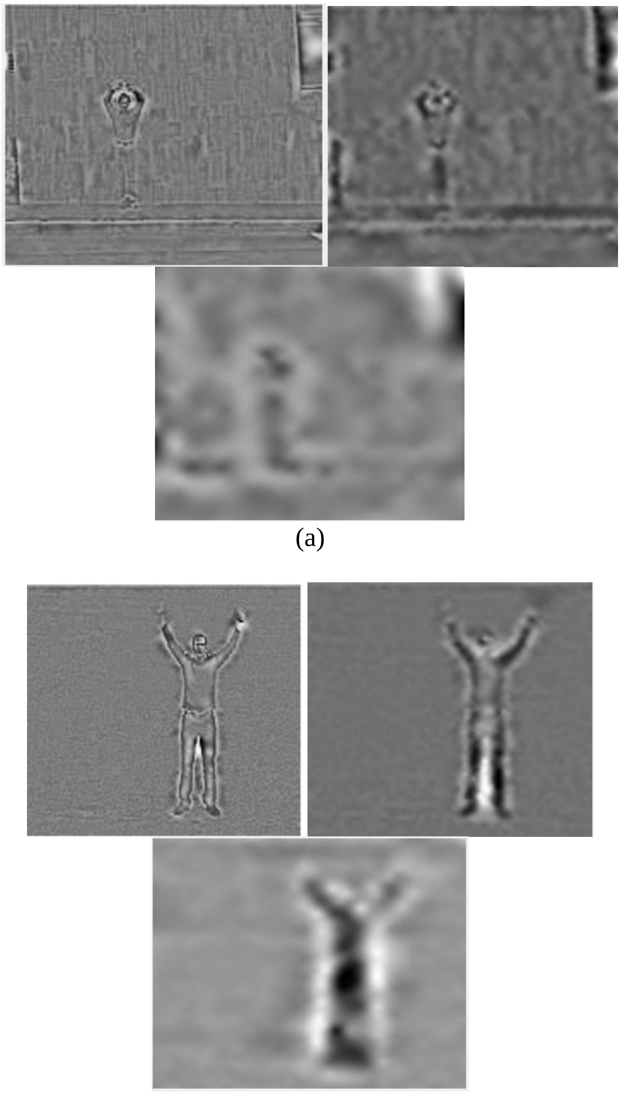
Step 9: This updated signal is used to obtain the residual signal, which is given in Eq. (9).

$$r_k = r_{k-1}(x,y) - bimf_k(x,y) \quad (9)$$

Step 10: Steps 1 and 9 are repeated for k^{th} items. $K = (k+1)$ when the residual $r_K(x,y)$ is monotonic signal. The BEMD decomposition process is then stopped. The real image can be expressed as the sum of all BIMF and residual r_K. This sum is given by Eq. (10).

$$f(x,y) = \sum_{k=1}^{K} bimf_k\,(x,y) + r_k(x,y) \quad (10)$$

Fig. 3 (a) and (b) shows feature extracted image of IMF1, IMF2 and IMF3 using BEMD for Weizmann and KTH datasets.



(a)

(b)

Figure.3 (a) Feature extracted image of IMF2, IMF2, and IMF3 using BEMD for Weizmann dataset and (b) Feature extracted image of IMF2, IMF2, and IMF3 using BEMD for KTH dataset

3.1.4. Scale invariant feature transform

The major objective of SIFT is to find the locations and key points of the frame at which feature is invariant to scaling and rotation. SIFT algorithm consists of four stages: scale-space extrema detection, key point localization, orientation assignment, and key point descriptor. The scale space extrema detection is utilized to extract the multi-scale features of input frame data. The SSED can be significantly achieved using a scale space function, it is based on Gaussian function. The function of SSED is expressed in Eq. (11).

$$L\,(x,y\;\sigma) = G(x,y,\sigma) * I\,(x,y) \quad (11)$$

Here, $I\,(x,y)$ is input image, * is convolution operator, $G(x,y,\sigma)$ is variable denoting Gaussian function convolution kernel, it is described by the Eq. (12).

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{(x^2+y^2)}{2\sigma^2}} \quad (12)$$

The difference of Gaussians is calculated by detecting the key point locations, locating scale-space extrema, $D(x,y,\sigma)$, and computing the difference between two the images. It is expressed by Eq. (13).

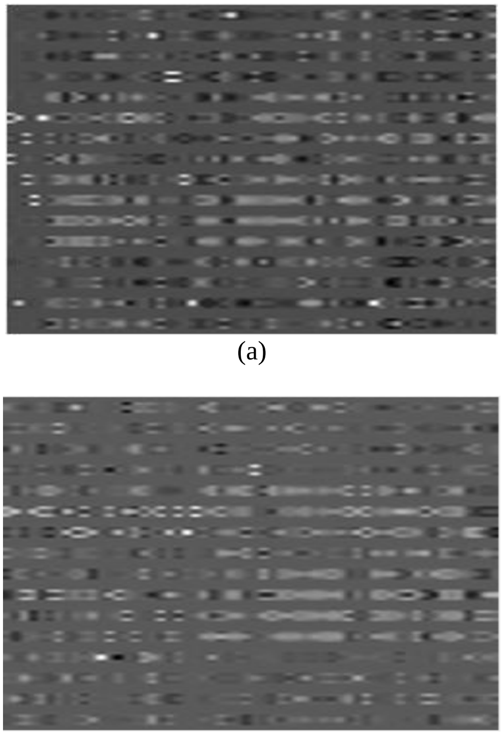
$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (13)$$

In key point localization, the key points are located by choosing the local extrema. In the model, the candidate points that are highly stable under the different of Gaussian space are represented as a key point. To find the orientation using the key points, select the Gaussian smoothed image (L), from the computing gradient magnitude(m), it is described by the Eq. (14) and (15).

$$m(x,y) = \sqrt{\begin{matrix}\big(L(x+1,y) - L(x-1,y)\big)^2 + \\ \big(L(x,y+1) - L(x,y-1)\big)^2\end{matrix}} \quad (14)$$

$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1-L(x,y-1))}{L(x+1,y)-L(x-1,y)}\right) \quad (15)$$

Once a key point is located, a descriptor is generated on the basis of the orientation, scale and location of the key points. Fig. 4 (a) and (b) feature extracted using SIFT in Weizmann and KTH dataset.



(a)

(b)

Figure.4 (a) Feature extracted using SIFT in Weizmann dataset and (b) Feature extracted using SIFT in KTH dataset

3.1.5. Discrete wavelet transform

The Wavelet Transform (WT) is applied to the individual slices to extract the mean, variance and standard deviation of the segments such that the performance of the classification is highly improved. As a result, four wavelets are generated in that one is a low-frequency wavelet and three are high-frequency wavelets. The wavelets are given by the Eq. (16).

$$\left[LL_{T_k}^{i,j}\ LH_{T_k}^{i,j}\ HL_{T_k}^{i,j}\ HH_{T_k}^{i,j}\right] = WT\left[T_k^{i,j}\right] \qquad (16)$$

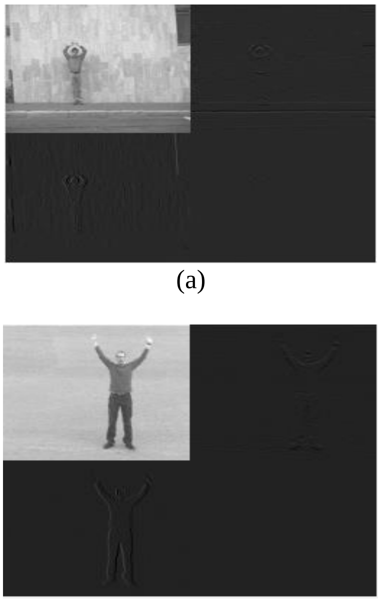
Here, $LL_{T_k}^{i,j}$,$LH_{T_k}^{i,j}$, $HL_{T_k}^{i,j}$, $HH_{T_k}^{i,j}$ and $T_k^{i,j}$ are the wavelets obtained by applying the WT to the j^{th} segment of the i^{th} modality. The high frequency wavelets $LH_{T_k}^{i,j}$ and , $HL_{T_k}^{i,j}$ are subjected to feature extraction as given in Eq. (17), (18), (19) and (20).

$$[\ \varepsilon_k^{LH}\ \ \mu_k^{LH}\ \ V\varepsilon_k^{LH}\ \ K_k^{LH}\ N_k^{LH}] = f\left[LH_k^{i,j}\right] \qquad (17)$$

$$[\ \varepsilon_k^{HL}\ \ \mu_k^{HL}\ \ V\varepsilon_k^{HL}\ \ K_k^{HL}\ N_k^{HL}] = f\left[HL_k^{i,j}\right] \qquad (18)$$

$$f^3 = \left[\ \varepsilon_{T_k}^{LH}\ \ \mu_{T_k}^{LH}\ \ V\varepsilon_{T_K}^{LH}\ \ K_{T_k}^{LH}\ N_{T_k}^{LH}\right] \qquad (19)$$

$$f^4 = \left[\ \varepsilon_{T_k}^{HL}\ \ \mu_{T_k}^{HL}\ \ V\varepsilon_{T_K}^{HL}\ \ K_{T_k}^{HL}\ N_{T_k}^{HL}\right] \qquad (20)$$



(a)

(b)

Figure.5 (a) Feature extracted using DWT in Weizmann and (b) Feature extracted using DWT in KTH dataset

Here, f^3 and f^4 are the feature vectors obtained from the j^{th} segment using wavelet transform in LH and HL frequency bands, which are of dimension $[1 \times 5]$. Hence, the feature obtained from the individual segment of the modality is calculated by the Eq. (21).

$$F_k^{i,j} = [f^1\|f^2\|f^3\|f^4\|] \qquad (21)$$

Fig. 5 (a) and (b) shows the feature extracted using DWT in Weizmann and KTH dataset

Finally, in the training process, the output of the three feature extraction is considered a hybrid feature extracted value, which is used as a testing data. In the training process, the input frame is pre-processed and features are extracted from the frame by using BEMD, SIFT and DWT. At the final stage, the trained and tested data are given to CNN for recognition of actions in the video.

3.2 Classification using CNN method

In this work, the training and testing data are given to CNN classifier for classify action recognition. In this research, the CNN classifier is established based on the concept of the Hubel and Wiesel study in Cat Visual Cortex (CVC). In this classifiers, the neurons consider as small portion of the frame that is known as sub-frames. Then, the respective sub-frames are utilized for feature extraction, for instance, a feature may be vertical-line and circle. Feature are captured by respective feature map of the network. Combination of features are classified by frames. Additionally, different feature maps are used to create the network robust to vary level of noise, colour and contrast and brightness. In CNN classifier, the convolution layer is a primary layer, which extracts the local information of frame: shape, texture and other features. Furthermore, the convolution operation enhances the input features and minimize the noise interference. Mapping operation in convolution process is mathematically represented in the Eq. (22).

$$x_j^i = f_c\left(\sum_{i \in Mj} x_i^{l-1} \times k_{i,j}^l + \theta_j^i\right) \tag{22}$$

Where, x_j^l represented as the j^{th} mapping set of convolution layer l, x_i^{l-1} represented as the i^{th} feature set denoting in the $(l-1)$ convolution layer and $k_{i,j}^l$ represented as the convolutional kernel between i^{th} feature set and j^{th} mapping set in convolutional layer l. Variable θ_j^l represented as bias and f_c denoted as activation function. Next step is pooling process; it reduces the possibility of over fitting during training process. The pooling process is expressed in Eq. (23).

$$x_j^l = f_p\left(\beta_j^l down\left(x_i^{l-1}\right) + \theta_j^l\right) \tag{23}$$

Here, $dowm\,(.)$ denoted as the down sampling approach from layer $(l-1)$ to layer l^{th}, θ_j^l and β_j^l are represented as the additive bias and multiplicative bias and $f_p(.)$ represented as the activation function. Commonly, the pooling process is subdivided into two types like maximum and average pooling. Finally, the pooling layers are arranged to form a rasterization layer. It is further associated with fully associated layer. The output of node j is mathematically denoted in Eq. (24).

$$h_j = f_h\left(\sum_{i=0}^{n=1} w_{i,j}\, x_i - \theta_j\right) \tag{24}$$

Here $w_{i,j}$ denoted as the association weight of input vector x_i and θ_j is represented as the node threshold and $f_h(.)$ is activation function. If the input layer deals with multiclass problem, softmax classifier is used in fully associated layer. Loss function of softmax classifier is represented in Eq. (25).

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} l\{y^{(i)} = j\}\, log\, \frac{e^{\theta_j^l}}{\sum_k \theta_k^l}\right] \tag{25}$$

Here, θ_j^l is input of j^{th} neuron in l layer, $\sum_k \theta_k^l$ is input of all neurons, $\frac{e^{\theta_j^l}}{\sum_k \theta_k^l}$ is output of j^{th} neuron, e is constant and $l(.)$ is indictor function. If the value in brace is true, output of the indictor function is zero. Next, add the rule items in $J(\theta)$ to prevent from falling into local optimum. Loss function of softmax classifier $J(\theta)$ after adding the rule items is mathematically denoted in Eq. (26).

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} l\{y^{(i)} = j\}\, log\, \frac{e^{\theta_j^l}}{\sum_k \theta_k^l}\right] + \frac{\rho}{2}\sum_{i=1}^{k}\sum_{j=0}^{n} \theta_{i.j}^2 \tag{26}$$

Here $\frac{\rho}{2}\sum_{i=1}^{k}\sum_{j=0}^{n} \theta_{i.j}^2$ is weighted term that helps to stabilize the excessive parameters in training set.

4. Result and discussion

This section is describe evaluated the proposed method. First introduce datasets along with experimental protocol and evaluations metrics. Then provide the implementation details of proposed HFE-CNN-VAR method. In this section conduct many experiments on the standard datasets and compare with baseline methods and number of recent techniques to demonstrate the effectiveness of the proposed HFE-CNN-VAR method. The HFE-CNN-VAR method was validated by three databases such as Weizmann dataset, KTH database and own database.

4.1 Weizmann database

The Weizmann dataset includes 10 human actions performed by 9 people. The videos are recorded from a static viewpoint. Different actions available in Weizmann database such as walking, running, jumping, skipping, bending, Pjumb (jumping in place of two legs), jack (jumping jack), side (galloping sideways), skipping, wave 1 (waving one hand), wave 2 (waving two hands). The video sequences are captured in simple background environment with a frame rate of 5fps. The resolution of the video sequence is 180 × 144 pixels. The dataset delivers a collection of a large number of actions suitable for analysing the accuracy of HFE-CNN-VAR method for action recognition. This Weizmann database includes approximately 5687 frames and a total of 93 video sequences. Fig. 6 shows the samples frame from Weizmann database.

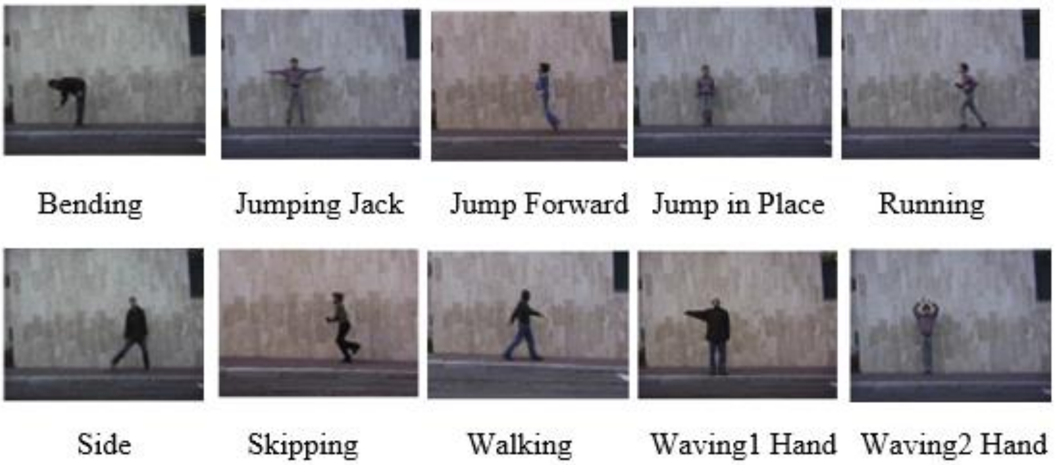


Figure.6 Sample frame from Weizmann database

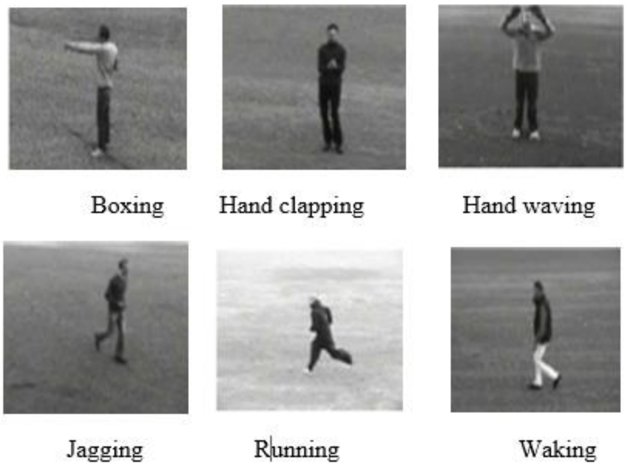


Figure.7 Sample frame from KTH database

4.2 KTH database

The KTH database consists of actions such as boxing, hand clapping, jogging, waving, running and walking. Each action was performed once by twenty-five subjects in 4-various scenarios, respectively. This 4-scenarios include outdoors (S1), outdoors with scale variations (S2) and outdoors with scale variations (S2), outdoors with different clothes (S3) and indoor (S4). Fig. 7 shows the sample frame from KTH database.

4.3 Own database

This dataset contains 300 video sequences captured using three different cameras with different resolutions and different orientations where ten users performing ten different actions: walking, running, skipping, one hand-waving, two-hand waving, jacking, jumping in place, jumping jack, gallop sideways and bend. The actions are recorded with reference to the Weizmann dataset but background objects are also considered in while recording own dataset. Fig. 8, 9 and 10 show long, rotate and short view of sample frame from own dataset.

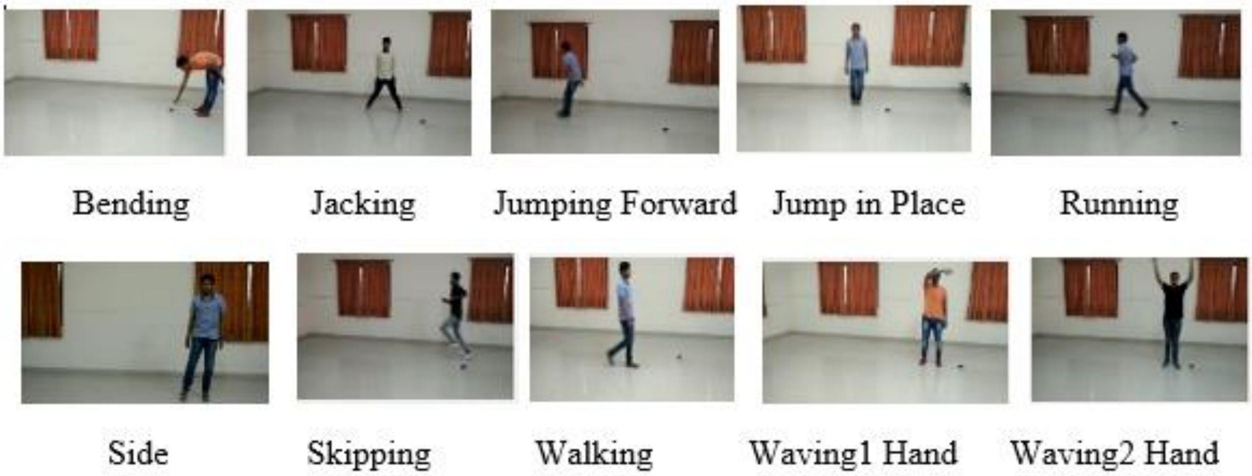


Figure.8 Long view of sample frame from own dataset

Figure.9 Rotate view of sample frame from own dataset

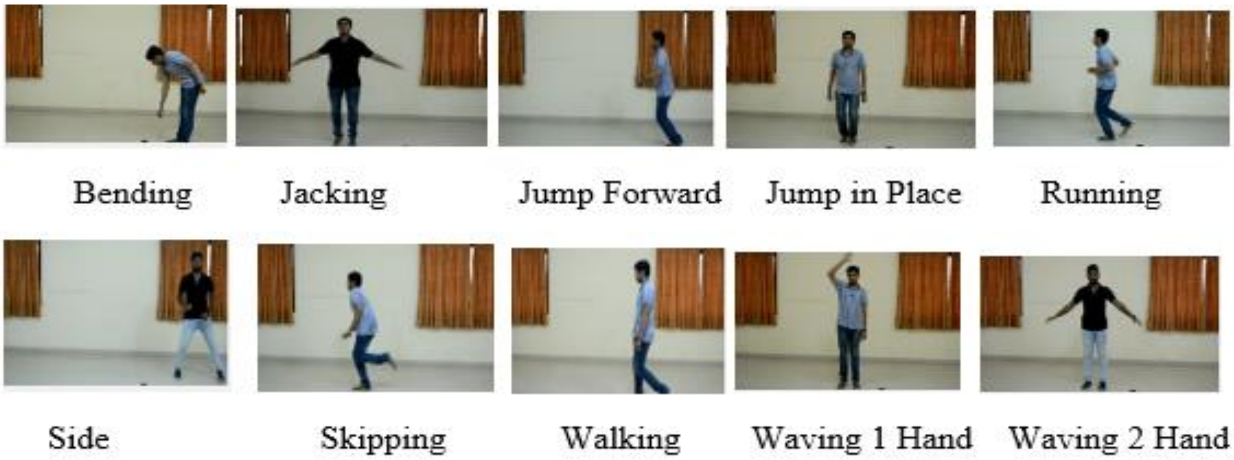


Figure.10 Short view of sample frame from own dataset

4.4 Performance analysis of the proposed HFE-CNN-VAR method on different databases

Tables 1, 2 and 3 provide the performance analysis such sensitivity, specificity and precision for different activities in Weizmann, KTH and own databases. The tabulated outputs prove that the proposed HFE-CNN-VAR method is much robust in terms of activity recognition with 99.33 % accuracy for the Weizmann dataset. The HFE-CNN-VAR correctly recognizes all activities in Weizmann database. The performance of the proposed method is particularly better in terms of various performance measures owing to the robust structural and feature extracted through the improved representation.

The proposed HFE-CNN-VAR method improves the classification results for activity classes in the KTH database. In the KTH database, the activities such as running, jogging, walking and hand waving are properly recognized with an accuracy rate of 99.01 %. The similar activities that share large similarities in shape and motion: boxing, jogging, running, walking, are very well distinguished using the proposed method. This misclassification of hand clapping and hand waving are considered as a false negative in the hand action category and a false positive in the hand clapping category. Boxing, jogging, walking and running are well distinguished using the proposed method.

Table 1. Performance analysis of VAR-HFE-CNN method for Weizmann database

Activities	Sensitivity (%)	Specificity (%)	Precision (%)
Bend	100	100	100
Jack	100	100	97
Jump Forward	97	100	100
Jump in Place	100	100	100
Run	100	100	100
Side	100	100	100
Skip	100	100	100
Walk	100	100	100
Wave 1 hand	100	100	97
Wave 2 hand	97	100	100

Table 2. Performance analysis of VAR-HFE-CNN method for KTH database

Activities	Sensitivity (%)	Specificity (%)	Precision (%)
Boxing	100	100	100
Handclapping	100	99	97
Hand waving	97	100	100
Jogging	100	100	100
Running	100	100	100
Walking	100	100	100

Table 3. Performance analysis of VAR-HFE-CNN method for own database

Activities	Sensitivity (%)	Specificity (%)	Precision (%)
Bend	100	100	97
Jack	100	100	97
Jump Forward	97	94	66
Jump in Place	100	95	70
Run	40	100	100
Side	90	100	100
Skip	100	100	100
Walk	100	100	100
Wave 1 hand	79	100	96
Wave 2 hand	90	100	100

Table 3 shows the performance analysis of HFE-CNN-VAR method for own database. Here only two activities correctly predicted by using proposed method.

Table 4 shows the performance of the HFE-CNN-VAR method for different datasets. The proposed method was obtained 99.33 % of accuracy in the Weizmann database with 10 activities (bend, jack, jump forward, jump in place, run, side, skip, walk1, wave1, and wave2), 99.01 % of the accuracy in KTH database with 6 activities (boxing, handclapping, hand waving, jogging, running and walking) and 90 % of accuracy in own database with 10 activities by three different camera views: long view, rotate view and short view. In Tab.4, the accuracy of recognition 99.33% is obtained by proposed method in Weismann datasets. So, the proposed method is mostly accurately predicted the human actions in Weismann dataset compared to other two datasets.

Table 5 shows the comparison of average Recognition accuracy of state-of-the-art and proposed method recognition accuracies (%) for the Weizmann, KTH and Own database. The performance of HFE-CNN-VAR method measured by utilizing efficient evaluation metric such accuracy with respect to different action recognitions: Bend, Jack, Jump Forward, Jump in Place, Run, Side, Skip, Walk and Wave 1 hand. The proposed HFE-CNN-VAR method shows better results compared to exiting methods: Zhang et al. [13], Yao et al. [14] and Chou et al. [18]. In Zhang et al. [16], this proposed method was obtained 96.66 % and 94.9 % of accuracy in Weizmann and KTH database. L. Yao, Y. Liu, and S. Huang [17], proposed a new method which obtained 95.83 % of accuracy in the KTH dataset. Chou [18], this proposed method was obtained 95.56 % and 90.58 % of accuracies in Weizmann and KTH databases. The proposed method has obtained better results in three different databases compared to existing action recognition methods, 99.33 % of accuracy in Weizmann database, 99.01 % of accuracy in KTH database, and 90 % of accuracy in own dataset. In this research, initially the frames were extracted from input videos and the frames are resized in the pre-processing. After the pre-processing, the object detection was done by Blob detection algorithm and tracking the object frame by frame was done using Kalman filter. Then the features were extracted from the moving object using feature extraction algorithms such as BEMD, SIFT and DWT. These feature extraction techniques were applied on pre-processed frames to extract the efficient features from multi-scale images. Finally, the features are given to CNN classifier for action recognition prediction. Generally, when the video is very poor quality due to motion blur, the proposed method helps to generate effective representation for video frames through extract feature identification process. In this work, the result analysis in proposed method shows that the most of the activities has yield better results compared to existing performance analysis.

Table 4. Accuracy (%) of the VAR-HFE-CNN method for different datasets

Proposed method	Number of activities	Database	Accuracy of Recognition Rate (%)
HFE-CNN-VAR	10	Weizmann	99.33
	6	KTH	99.01
	10	Own data set	90

Table 5. State-of-the-art and proposed method recognition accuracies (%) for the Weizmann, KTH and Own database

Author	Methods used	Weizmann Database	KTH Database	Own Database
Zhang et al. [13]	SIFT, BoW, SVM	96.66 %	94.9 %	-
Yao et al. [14]	STBi-fusion + POOL STRep	-	95.83%	-
Chou et al. [15]	Nearest Neighbor, Gaussian Mixture Model, Nearest Mean Classifier	95.56 %	90.58%	-
Proposed HFE-CNN-VAR method	BEMD, SIFT, DWT, CNN	99.33%	99.01%	90 %

5. Conclusion

This paper presents an approach for human action detection, tracking and action recognition in multiple datasets. The Blob detection and Kalman filtering approaches were used for detection and tracking of human in the video sequences. The view-invariant features were obtained by extracting holistic features from different feature extraction techniques such as BEMD, SIFT and DWT to address multi-view action recognition from various perspectives for appropriate and robust action recognition under the different changes. The obtained CNN classifier is able to recognize fine-grained actions from different videos. The experiments results were evaluated on the KTH, Weizmann and own datasets. The proposed HFE-CNN-VAR method was achieved 99.33% of accuracy in Weizmann dataset, 99.10% of accuracy in KTH dataset and 90% of accuracy in own dataset compared to existing methods. In future work, the optimization algorithm with object tracking will be implemented to reduce the time complexity of the action recognition system.

References

[1] M. Ma, N. Marturi, Y. Li, A. Leonardis, and R. Stolkin, "Region-sequence based six-stream CNN features for general and fine-grained human action recognition in videos", *Pattern Recognition*, Vol.76, pp.506-521, 2018.

[2] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos, "Robot learning manipulation action plans by Watching unconstrained videos from the world wide web", *In Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[3] Z.A. Khan and W. Sohn, "Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care", *IEEE Transactions on Consumer Electronics*, Vol.57, No.4, pp.1843-1850, 2011.

[4] Y. Xiao, J. Chen, Y. Wang, Z. Cao, J.T. Zhou, and X. Bai, "Action recognition for depth video using multi-view dynamic images", *Information Sciences*, Vol.480, pp.287-304, 2019.

[5] L. Chen, S. Zhanjie, J. Lu, and J. Zhou, "Learning principal orientations and residual descriptor for action recognition", *Pattern Recognition*, Vol.86, pp.14-26, 2019.

[6] A. Saleh, M. Abdel-Nasser, M. Angel Garcia, and D. Puig, "Aggregating the temporal coherent descriptors in videos using multiple learning kernel for action recognition", *Pattern Recognition Letters*, Vol.105, pp.4-12, 2018.

[7] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition", *IEEE Transactions on Image Processing*, Vol.24, No.3, pp.980-993, 2015.

[8] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.35, No.1, pp. 221-231, 2013.

[9] Y. Han, P. Zhang, T. Zhuo, W. Huang, and Y. Zhang, "Going deeper with two-stream ConvNets for action recognition in video surveillance", *Pattern Recognition Letters*, Vol.107, pp.83-90, 2018.

[10] F. He, F. Liu, R. Yao, and G. Lin, "Local fusion networks with chained residual pooling for video action recognition", *Image and Vision Computing*, Vol.81, pp.34-41, 2019.

[11] Y. Zhao, H. Di, J. Zhang, Y. Lu, F. Lv, and Y. Li, "Region-based Mixture Models for human action recognition in low-resolution videos", *Neurocomputing*, Vol.247, pp.1-15, 2017.

[12] C. Zhang, Y. Tian, X. Guo, and J. Liu, "DAAL: Deep activation-based attribute learning for action recognition in depth videos", *Computer Vision and Image Understanding*, Vol.167, pp.37-49, 2018.

[13] J. Zhang, Y. Han, J. Tang, Q. Hu, and J. Jiang, "Semi-supervised image-to-video adaptation for

video action recognition", *IEEE Transactions on Cybernetics*, Vol.47, No.4, pp.960-973, 2017.

[14] L. Yao, Y. Liu, and S. Huang, "Spatio-temporal information for human action recognition", *EURASIP Journal on Image and Video Processing*, Vol.39, No.1, 2016.

[15] K.P. Chou, M. Prasad, D. Wu, N. Sharma, D. Li, Y. Lin, M. Blumenstein, W. Lin, and C. Lin, "Robust Feature-Based Automated Multi-View Human Action Recognition System", *IEEE Access*, Vol.6, pp.15283-15296, 2018.